SUPERBOT
CHALLENGE

PRESENTED BY
neumont
UNIVERSITY

SUPERBOT CHALLENGE™
GREENFOOT PROGRAMMING CONTEST

INSTRUCTOR'S PACKET

SUBMISSION DEADLINE: MARCH 20, 2010

www.superbotchallenge.com

IMPORTANT CONTACT INFORMATION

Area Coordinator: Nate Blanchard
Phone: 801-302-2873
Email: superbotchallenge@neumont.edu

Welcome to the SuperBot Challenge.

This SuperBot Challenge programming contest is designed to provide high school seniors and juniors (sophomores and freshmen are welcome to compete as well) with an opportunity to test their Java programming skills. The Challenge provides students an opportunity to explore Object Oriented Programming in a fun, hands-on format, while competing for scholarships and prizes.

The SuperBot Challenge uses Greenfoot, Open Source software created by the University of Kent in Canterbury, UK. Programming SuperBot in the Greenfoot IDE should provide high school students an opportunity to stretch their Java programming skills and prepare for advanced courses (or college level study) in computer science.

Enjoy!

I. CHALLENGE DESCRIPTION:

The SuperBot Challenge provides beginning, intermediate, and advanced computer science students an opportunity to explore Object Oriented Programming. SuperBot uses the Java Programming Language in the Greenfoot IDE.

Required Materials:

All SuperBot Challenge scenarios, instructional materials, and Greenfoot software are available online at: **www.superbotchallenge.com**.

- To run Greenfoot, competitors must have Java 5 or Java 6 (JDK 6) installed. For Windows Vista, Java 6 is required
- A Greenfoot Installation
- The SuperBot Challenge Scenario
- An Internet connection and the ability to zip a file and post it to the SuperBot Entry submission page

Available Supplementary Materials:

- A SuperBot Challenge lesson plan for classroom use (following pages)
- Greenfoot Tutorial Videos (available on disc and web)
- A Greenfoot discussion forum for students
- The Greenroom - an instructor's discussion board
- Additional learning scenarios for classroom use
- Links to the Greenfoot technical support website
- Classroom presentations (limited availability)
- Loaner Greenfoot Instructional Manuals (upon request, limited availability)

II. CHALLENGE OVERVIEW

Ganthum city is overwhelmed by an outbreak of crime. Sticky Fingers McKrackin and his evil mob of bandits are on a crime spree, robbing the banks and small businesses that dot the streets of this otherwise unhurried and unworried community. Only SuperBot can save them!

Program SuperBot to collect all the loot abandoned by these marauding villains. But watch out! Sticky Fingers and his evil mob aren't going to give up their loot without a fight! As SuperBot collects the bags of loot in one part of town, Sticks LaFleur will be chasing behind, picking up abandoned loots and hiding them.

Hurry! Teach SuperBot to maneuver through the streets of Ganthum, avoiding Sticky Fingers, buildings, and wayward teens in order to collect the stolen loot and return it to the Bank. Program SuperBot carefully so that he can succeed in returning all the money to the Bank before Sticky Fingers gets his hands on it.

III. CHALLENGE OBJECTIVES:

Greenfoot is an excellent vehicle to introduce the concepts of Object Oriented Programming in the classroom. It allows students to explore basic and advanced programming concepts in a safe, hands-on format. SuperBot requires less faculty supervision than a more complex programming environment. Superbot is an intuitive introduction to Java and the basic concepts students will master in later computer science courses. By using the pre-constructed SuperBot classes even beginning students may experience SuperBot Challenge success with little classroom instruction. Additionally, SuperBot's grid system is an excellent introduction to the AP Computer Science curriculum while still supporting the educational needs of lower level computer science courses.

Through the SuperBot Challenge students will:

1. Be introduced to programming in an object-oriented language
2. Instantiate objects and call methods to change the state of that object or act with a desired behavior.
3. Write methods that extend the object's pre-existing behaviors
4. Design an algorithm for collecting objects

IV. INSTRUCTION METHODS:

SuperBot Challenge is designed for classroom delivery, but also works well for independent study at home. Students should begin with the basic Greenfoot Tutorial videos and run the current SuperBot scenario before beginning to program their Bot. Neumont has provided a pre-coded character "StickyFingers" for use as an instructional aide. By reading StickyFingers' code, students with a basic understanding of Java should be able to create a strategy and search logic for their own Bot. It may be a good idea to open "StickyFingers" as a group and discuss the code with your students. Once the students have programmed they should race them against StickyFingers to determine if their search logic is complete and competitive.

V. EXPLORING SUPERBOT IN THE CLASSROOM

TEACHING SUPERBOT	AREAS TO COVER
Day 1	Introduce your students to the SuperBot Challenge and take them to the SuperBot website. Through the website students can load Greenfoot and the SuperBot Challenge scenarios onto classroom computers.
Day 2	Watch the tutorial videos, and walk the students through the existing code for “GanthumCity” and “StickyFingers.” Allow students to work along.
Day 3	Allow students to experiment with programming SuperBot in a minimally supervised learning session. On day Three, students should familiarize themselves with Greenfoot by creating their own SuperBot World.
Self Study	After day three, students should feel confident coding SuperBot on their own. Greenfoot's simple download process, and the compact nature of the Challenge files make SuperBot easily portable from the classroom to home.
Final Project	Students submit their finished projects online. See the “Submit Your Entry” page on the Neumont website for submission details.

VI. CHALLENGE REQUIREMENTS

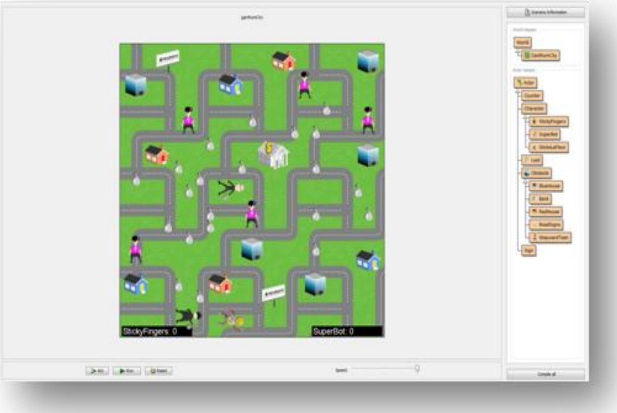
Basic Requirements:

1. All work must belong to the competitor.
2. Challengers are allowed to use online resources for writing SuperBot code, but the code should not be "cut and pasted" from the code of another competitor or website.
3. All projects must be submitted through the SuperBot submission page as a zip file on or before March 20, 2010. ***Late entries may not be accepted.***
4. Scholarship competitors will be notified of eligibility after the quarter-final round of the Challenge. Please update Neumont University with any changes to contact information.
5. All "Bots" must be capable of running on a combination of courses against Sticky Fingers, a Neumont Bot, and the Bot of another competitor.
6. All "Bots" must be self-propelled when "run" is pressed.
7. "Bots" will not be driven during the competition.

All Projects Must Include:

1. A fully programmed SuperBot renamed as directed in "Naming the Bot"
2. A map created by the competitor and saved in “World.”
3. Challengers should update their “World” using "populate" under “World”.
4. Challengers should then populate “World” with the correct name for the newly programmed Bot.
5. Competitors should not add new images, characters, or obstacles to their World. They should only relocate "loots" and obstacles.

SuperBot Rules:



1. SuperBot has Super Powers. He can leap buildings in a single bound, move a building's location, or devise new and creative ways to collect the "loot" -- it's up to each competitor. See "Programming SuperBot" for limitations.
2. SuperBot may not destroy obstacles or opponents, he is a peaceful "Bot".
3. SuperBot may only collect and return "loot" to the bank. Obstacles/opponents may be moved but not destroyed.
4. SuperBot will encounter a variety of obstacles. It is best to build "SuperBot" with rational search logic rather than a specific map memory. Maps, obstacles, and Bots SuperBot encounters will change throughout the competition.
5. All competitor created maps may only use obstacles provided in the current SuperBot World. Characteristics of these obstacles may not change.

Naming the Bot:

1. Each Challenger should name their Bot with the Challenger's name, including first initial, and the first five letters of the Challenger's last name, followed by Bot. Ex. *Bruce Wayne's Bot would be named: bwaynebot*
2. Challengers with a last name shorter than five characters should use their whole last name, Ex. *Bat Man's Bot would be named: bmanbot.*

General Programming Rules:

1. Challengers may use methods in Character and Obstacle as well as methods that contestants write in the SuperBot class.
2. Challengers cannot use attributes from the Character or the Obstacle class unless they are public.
3. Challengers cannot use methods or attributes in the Actor class unless they appear in the Character class.
4. Challengers cannot use methods or attributes in the World class.
5. Challengers may use methods in Java's Math class.
6. Challengers should not hardcode any numbers except the numbers -1, 0, and 1.
7. Challengers may not import anything other than Greenfoot.
8. Challengers cannot change any code provided except for inside SuperBot and the competitor's World.

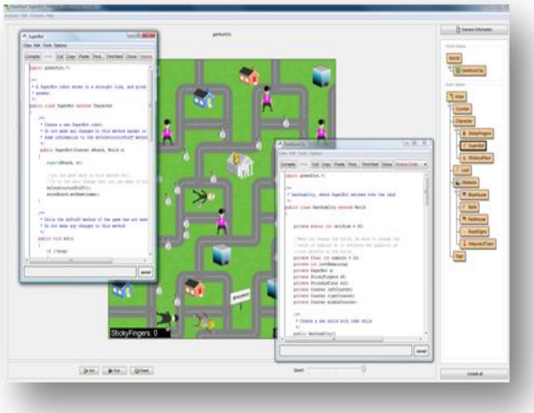
Programming SuperBot:

1. SuperBot may turn as many times as needed within an act method.
2. SuperBot cannot walk on top of obstacles, but he may jump, move, dodge and/or destroy obstacles. SuperBot may walk on streets or grass. Get creative, but be careful. Whatever makes the game easier for one Bot may also help an opponent.
3. SuperBot's name (not the class name but the String called name) must be changed.

Scoring in SuperBot:

1. Each character gets one point for picking up a bag of loot.
2. In order to pick up the loot, SuperBot must be directly on top of the loot.
3. When a Bot brushes past the loot and does not appear to pick the loot up the Bot and the loot were not actually in the same cell.
4. Each character gets five points for dropping a bag of loot in the bank.
5. SuperBot must physically deliver the loot to the bank to make a deposit; he may not throw the loot at the bank or magically transport the loot.
6. The loot must be within the image of the bank after a deposit is made (not outside it) or a score will not register.
7. If the counter does not register a loot deposit the loot was deposited into a cell near, but not at the bank.
8. SuperBot and StickyFingers cannot pick up loot that has been delivered to the bank.
9. A character can call the move or transport methods just once within each act method, except when putting loot in the bank.
10. When putting loot in the bank, a Bot can move or transport into the bank, drop the loot, then move or transport out of the bank all within one act method.
11. SuperBot may not to steal loot from the bank. SuperBot is an honest Bot.
12. SuperBot may not drop the loot and pick up again to inflate the score.
13. SuperBot's goal is to return all loots to the bank, before StickyFingers snatches it up and takes it to his hideout.

Designing a New World:



1. Once SuperBot is programmed design a World that is best suited to the search logic created.
2. Only use provided images and obstacles.
3. Do not change the characteristics of the obstacles provided.
4. When changing the world, be sure to change the value of "numLoot" so it reflects the quantity of loot objects in the world.
5. Code GanthumCity with an odd number of Loots to prevent a tie.
6. Check that loot is not located directly over any obstacles or on the bank -- this will prevent the Bots from clearing the screen and result in a null game.
7. Uncomment the following line of code to run a new world "contestantPopulate()"

Challengers may relocate only the following items in their World:

LOOT

HOUSES

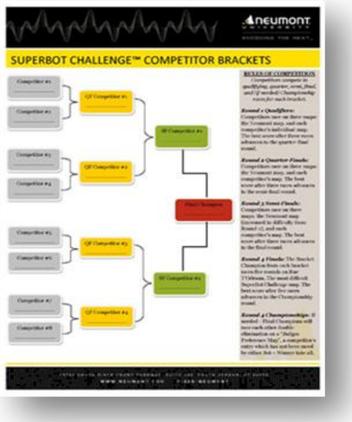
WAYWARD TEENS

BUILDINGS

THE BANK

ROAD SIGNS

VII. JUDGING SUPERBOT:


SUPERBOT CHALLENGE™ COMPETITOR BRACKETS

SuperBot Challenge entries will be evaluated as follows:

1. **Components of Challenge Met:** (Map, Code is correctly compiled, SuperBot is named). Bots that do not meet basic Challenge criteria may not be allowed to compete. If an incomplete Bot is received before the final May 20th deadline we will attempt to notify the participant by email.

2. **Races:** Challengers compete in qualifying, quarter, semi, final, and (if needed) Championship races for each bracket. Competitors are placed into brackets at random and race according to their bracket position.

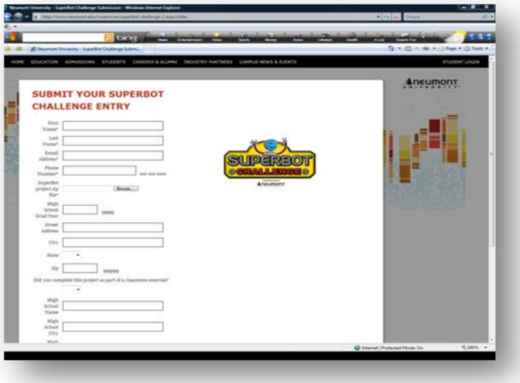
Rules of Play	
Qualifying Rounds	Competitors race on three maps: the Neumont map and each competitor's individual map. The best score (most points) after three races advances to the quarter-final round.
Quarter-Final Rounds	Competitors race on three maps: the Neumont map and each competitor's map. The best score after three races advances to the semi-final round.
Semi-Final Rounds	Competitors race on three maps: the Neumont map (increased in difficulty from the quarter-final round) and each competitor's map. The best score after three races advances to the final round.
Final Rounds	The Bracket Champion from each bracket races five rounds on Rue T'Orleans, the most difficult SuperBot Challenge map. The best score after five races advances to the Championship round.
Championship Rounds	If needed - Final Champions will race each other double elimination on a "Judges Preference Map", a competitor's entry which has not been raced by either Bot – Winner takes all.

VIII. TROUBLESHOOTING:

The SuperBot Challenge presented by Neumont University is available to handle limited technical support requests and clarifying questions about the rules of the Challenge. Additionally, Greenfoot producers host a Discussion Board on the usage and programming of Greenfoot scenarios, and provide general technical support by email.

Links and technical support information is available on the SuperBot website and directly through Greenfoot.

IX. SUBMITTING A SUPERBOT ENTRY



SUBMIT YOUR SUPERBOT CHALLENGE ENTRY

First Name*

Last Name*

Email Address*

Phone Number* ###-###-####

SuperBot project zip file*

High School Grad Year 9999

Street Address

City

State

Zip 99999

Did you complete this project as part of a classroom exercise?

High School Name

High School City

Submit entries online through the SuperBot Challenge website.

The student must provide:

1. Basic contact information (name, address, phone)

2. High School information (so that we can judge and award Educator's Grants)

3. A zipped file of their SuperBot project complete with all images and compiled code.

4. We recommend the student zips their file, tests emailing it to another computer, and re-opens before submitting to the SuperBot Challenge.

5. ***Neumont University is not responsible for incomplete or incorrectly compiled SuperBot submissions.***

X. AWARDS

Student Awards:

- 1st Place - $500 Best Buy Gift Card
- 2nd Place - $250 Best Buy Gift Card
- 3rd Place - $100 Best Buy Gift Card

SuperBot Challenge Neumont University Scholarship: Up to a full tuition Neumont University scholarship may be awarded to SuperBot Challenge participants who show exceptional academic promise, a strong background in computer science, and potential to excel in Neumont's rigorous environment.

Educator's Awards for Promoting Technology in the Classroom: Educators who encourage Object Oriented Programming, and especially the SuperBot Challenge, in their classroom are eligible to win **$500 Best Buy Gift Card Classroom Improvement Grants** sponsored by Neumont University. The Gift Cards will be given to the three teachers who sponsor the largest number of SuperBot entries.

Interested teachers should register at www.neumont.edu/superbot and provide the name of the High School where they will be hosting SuperBot Challenge workshops in the classroom. A representative from Neumont University will follow up with you to discuss your SuperBot classroom workshops and check that you have adequate Student Instructional Packets and supplemental SuperBot Challenge materials.

Good luck with SuperBot Challenge!